A Dynamic K-means Algorithm for Searching Conserved Encrypted Data in a Cloud

Avinash C.
Department of Computer Science
VIT University
Vellore – 15

Hasritha P.
Department of Computer Science
VIT University
Vellore – 15

Abhinay J.
Department of Computer Science
VIT University
Vellore – 15

ABSTRACT
The owner of large amount of data chooses cloud facility to outsource their precious data. For the purpose of privacy the data owners encrypted their data before outsourcing. This encryption hides the connection between the documents. So this makes the ciphertext search critical. Caused by software/hardware failure, data search results returning to the users may carry injured data. Thus, a verifiable method should be provided for users to verify the accuracy of the search results. This paper proposes a hierarchical clustering method in order to get a better clustering result. This method is based on k-means clustering algorithm. Every document will be clustered and hashed; this hash result will be used as the representative of the document. This paper gives a new search technique to adopt the backtracking algorithm on the above clustering method. With the increment of the data volume, the usefulness of the proposed method in rank privacy tends to be more evident. By appealing the Merkle hash tree and cryptographic signature to authenticated tree anatomy, we give a verification mechanism to persuade the accuracy and perfectness of query results. By using this method, the search time increased only sequentially not exponentially. The exploratory result demonstrates that the proposed system solves multi keyword search problems and also brings the relevance between retrieved documents and raises the search efficiency.

Keywords
Cloud computing, Ciphertext, Multi-keyword search, Hierarchical Clustering, Security, Query, Symmetric Key encryption.

1. INTRODUCTION
Nowadays, each and every field in a world wants to create a database to maintain their user's details and their profile. So the volume of data will be increased. For maintain all the data, another large database is needed. Cloud is the only solution to store large amount of data. It can reduce the storage management cost and storehouse efficiency expending. Some private data also wants to store in a cloud like passwords, health details, and secret information. But the cloud consumers and cloud server are not in the one particular territory; our outsourced data may be beneath the vulnerability of danger. To preserve the data privacy and encounter uninvited approach, hypersensitive data has to be encrypted before outsourcing. So the information will be very confidential in cloud and beyond the bounds. Nonetheless, ciphertext generates powerful data usage a very challenging task when cloud is a large amount of outsourced data files. And also the liaison between data is hid in the above technique. The liaison between the data means the properties of the data and hence retaining the relationship is very important to fully explicit the data. One document is grouped with another document by same category. This category makes a relationship between these documents. If a document is separated from the other documents except those are relevant to education, then it is accessible for users to affirm this document following the category of the education. As a result of encryption, this important liaison has been hiding in the traditional methods. Accordingly, introducing a technique which can retain and use this relationship to accelerate the search phase is desirable. If any software or hardware failure occurs, data hunt event returning to the users may contain injured data. So, a suitable method should be furnished for users to prove the accuracy and perfectness of the query results.

Searching of encrypted data over a cloud wants to be secure. So in traditional methods they used searchable encryption technique, that frame a searchable inverted index that stocks a list of mapping from keywords to the corresponding set of files which have this type of keyword. To retrieve all the information about the target document, initially the user gives the keyword as a input to the cloud, it creates the trapdoor regarding to the keyword and then this trapdoor is submitted to the cloud server. Then cloud server compares the trapdoor and the index, and then finally server gives all the files relevant to this keyword to user. But this process is only permit accurate single keyword search. This single keyword search method encrypts all the word in the document independently. The data searching cost will be increased cause of searching document word by word. Some another system adds inverted index with order preserving symmetric encryption; here relevant score is used to get the target files. This relevant score is encrypted by OPSE for security. This system increases the usability of system and reduces the communication overhead. This also based on single keyword ranked search. Then the fuzzy keyword search method also used to search a target document even the user enters mistook keywords. But this system is based on multi keyword search method, which means this system uses different keyword search technique over encrypted data. Using this efficiency of the searching process is increased. But here the relevance between the documents is hid by this encryption.

In this paper, each and every document is view as a point of high structural storage management. Based on the relationship between the documents, all documents are partitioned into several categories. And also, if any documents have minimum distance in the high structural storage management, these can be segregate into unambiguous category. So they can be easily identified. When discarding unrelated categories and electing the required category, the time used for searching will be reduced. In a large amount of data, the user only needs a little amount of data. So we create one more sub category from the

specific category because of the desired document which user aims at is very small. To retrieve the target document, backtracking algorithm is used in this paper. Based on this algorithm, the cloud server will first check the top categories and then enter into desired sub-category. From this sub category, the server gets the desired k documents related to the target document. If the sub category doesn't contain the k documents relevant to the target document, the server back into main category and retrieve the target document from its sibling categories. This process is continuous till the target document is securely retrieved from the cloud. Here this paper uses hash function to check the integrity of the target result. Every document stored in the database is hashed using verifiable structure. And the results of the hashed documents are used to identify these documents. To identify the present category of the target document, the hashed result of the k documents are again hashed with the category information about the target documents. So the consolidation of current category information and sub-category information denotes the category of the target document. Virtual root is the first level of hash result; the users can verify this root instead of verifying all the category information.

2. OUR CONTRIBUTION

To improve the search efficiency of our proposed system, we use hierarchical clustering index that retain the close relationship between the documents. This hierarchical clustering method assures the multi-keyword ranked search technique over ciphertext information. By this, even when the data volume is growing up the searching time is increased only linearly. Relevance score wants to be found between the keyword and documents, this relevance score is regard to the same specific field with the keyword. So the desirable documents noticed by the user only enough to evaluated to access their relevance score. To retain the relationship between the original documents in an encrypted storage this paper includes the clustering method. This method is grouping all documents which are containing the same properties among various documents in the dataset. For finding this relationship between documents, this method uses metric value called relevance score. The number of documents in the cluster and the relationship between the documents play a vital role in a clustering method. Because of restriction on the cluster will be broken when new document is added into this cluster. This time another new cluster center wants to be generated and the new document will be selected as a temporal cluster center. Later the entire cluster center will be re-established and all the documents will be reappointed. So the number of cluster depends on these two vital roles. These processes are done at dynamically. Our hierarchical method gives a better clustering result even it have large collection of data. The cluster size wants to be retained the threshold value between clustering correctness and query efficiency. In proposed method point of view, when the cluster maximal size is reduced the minimum relevance score and the number of clusters will be increased. Each and every cluster has the limitation in its size, if anyone surpasses this limitation again the cluster will partition into sub-cluster. Here the clustering process is finished after that we think about searching process, it is taken as a user phase. User gives a target document as a query to a cloud server, and then the cloud server will find the relevance score between cluster center and query then selects the closest cluster. The main aim of this action is finding correct query related cluster from the child cluster until that the process will be looped to earn the closest child cluster. If the cloud server did not find the accurate cluster the cloud server trace back into another cluster and find the accurate cluster. For further improvement, we also certify the integrity of the result using certified tree structure on the hierarchical clustering method. This structure has the profit of taking advantages on cryptographic technique and Merkle hash tree. Every document stored in the database is hashed using verifiable structure. And the results of the hashed documents are used to identify these documents. To identify the present category of the target document, the hashed result of the k documents are again hashed with the category information about the target documents. So the consolidation of current category information and sub-category information denotes the category of the target document. Virtual root is the first level of hash result; the users can verify this root instead of verifying all the category information.

For reference our contribution is summarized as follows:

1. For the first time we solve the problem of multi-keyword ranked search over encrypted file using hierarchical clustering index. Here the clustering algorithm retain the relationship between the desired documents
2. This proposed method is also suitable for large collection of data in cloud; here also the searching time is increased in linearly only not exponentially. To improve the rank privacy in this area, we use backtracking algorithm.
3. This structure has the profit of taking advantages on cryptographic technique and Merkle hash tree. This mechanism persuades the accuracy and perfectness of the query result.

The rest of this paper is organized as follows. In section III, we describe the system model, threat model, design goals, notations and preliminaries. Section IV provides the detailed description our proposed system. And in V section we define the security and efficiency of the proposed system. Section VI provides the concluding remark of the whole paper.

3. PROBLEM FORMULATION

3.1 System model

We examine the system model for the proposed system involving three distinct entities cloud server, data owner, data user. Data owner has large amount of data to store it in a cloud. When he stores the data into a cloud he wants to encrypt his document for the purpose of security. And also the cloud server needs to minimize the trouble of searching time to the data user. So the server clusters the document

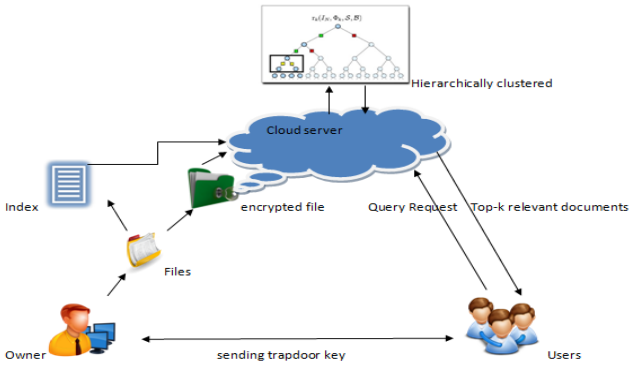

Fig 1. System model

using K-means clustering algorithm. The user wants to be authorized by the data owner for access the data. So initially

the user gets the access control from the data owner. After getting the access control, the user sends query request to the cloud server. The server verifies the authenticity of user, after that it search the query result in large database regarding to the data user query request. Then it returns the top-k documents to the user, and the user can access the relevant document based on his query. But the cloud server doesn't know about the content of the data stored by data owner. Here data owner, and data user only the trusted parties except cloud server. Cloud server is called as the semi-trusted party.

3.2 Threat model

The antagonist person's capability can be terminated in two threat models.

Known Ciphertext Model In this model, the encrypted documents are collected into one form and these collected documents are given to cloud server, and additionally the cloud server can get encrypted query keywords and encrypted data index.

Known Background Model Except that the above information, the cloud server can earn the additional information like statistical background information of dataset. By this he can analyze the specific keyword in the query. By these two ways the antagonist persons are identified.

3.3 Design Goals

Our scheme is designed to provide the accurate and perfect query result to the authenticated data user.

- Search efficiency. Even when the data volume is increased into large, our proposed model reduce the time complexity into logarithmic to search query document.
- Retrieval accuracy. Our proposed system has maintained the relationship between the query and the document in the result set. So the accuracy of returned document is high.
- Integrity. Accuracy, perfectness, and freshness these three categories improve the integrity of the search result. By this three properties, the server allow only authorized owner to upload the data and it didn't discard adequate data from the search result and also it verify all the documents uploaded by data owner are existed one or new one respectively.
- Privacy Requirements. Researchers focused on the following criteria.

1) Index Confidentiality. The values of keywords are collected in the index. Thus, the index stored in the cloud server wants to be encrypted;
2) Trapdoor Unlink ability. The cloud server could do some statistical analysis over the search result. Meanwhile, the same query should generate different trapdoors when searched twice. The cloud server should not be able to deduct the liaison between trapdoors.
3) Keyword Privacy. The cloud server could not detect the keyword in query, index by considering the statistical information like term frequency.
4) Rank Privacy. Rank order of inquiry events should be well guaranteed. If the rank order remains stable, the antagonist person can match the rank order of different search results, further determine the search keyword.
5) Data confidentiality. Data confidentiality presents the privacy, security, and retains the confidential format of documents. The opponent cannot acquire the plaintext of documents accumulated on the cloud server if data privacy is guaranteed. Symmetric cryptography is a conventional way to achieve data privacy. So most leading organization uses the Symmetric key encryption algorithm to prevent their privacy.

D. Notations and preliminaries

The notations and preliminaries used in this model are described as follows.

d_i	The i^{th} document vector, denoted as $d_i = \{d_{i,1}, \ldots, d_{i,n}\}$, where $d_{i,j}$ represents whether the j^{th} keyword in the dictionary appears in document d_i.
m	The number of documents in the data collection.
n	The size of dictionary D_W.
CCV	The collection of cluster centers vectors, denoted as $CCV = \{c_1, , c_n\}$, where c_i is the average vector of all document vectors in the cluster.
CCV_i	The collection of the i^{th} level cluster center vectors, denoted as $CCV_i = \{v_{i,1}, \ldots, v_{i,n}\}$ where V_i,j represents the j^{th} vector in the i^{th} level.
DC	The information of documents classification such as document id list of a certain cluster.
D_V	The collection of document vectors, denoted as $D_V = d_1, d_2, , d_m$.
D_W	The dictionary, denoted as $D_w = \{w_1, w_2, \ldots, w_n\}$.
F_w	The ranked id list of all documents according to their relevance to keyword w.
I_c	The clustering index which contains the encrypted vectors of cluster centers.
I_d	The traditional index which contains encrypted document vectors.
L_i	The minimum relevance score between different documents in the i^{th} level of a cluster.
QV	The query vector.
TH	A fixed maximum number of documents in a cluster.
T_w	The encrypted query vector for users query.

Fig 2 Notations

4. ALGORITHM AND DEFINITIONS OF BASIC SCHEME

4.1 System Formation

Multi-Keyword ranked search over encrypted data based on hierarchical clustering index is a proposed scheme in our project. It introduce the vector space model, here each and every document is denoted as a vector which means every document can be view as a point in a high structural storage management. These documents are arranged by a vector. The keyword is created from the vector dimensions and the values symbolizes whether the keyword presents in the document or not. Likewise, the query given by the user also represented as a vector. The cloud server estimates the inner product of the document vectors and query vector to find the relevance score between the query and desired documents. Finally, based on this top-k relevance score, the cloud server gives the target document to the user. And also to retain the relationship between the original documents over the encrypted document,

the index vector of their documents are clustered based on clustering algorithm. If the documents having high relevance score, they can be easily clustered for further process. So by this hierarchical clustering method, server search a document step by step groups instead of search whole database.

4.2 Background on MRSE-HCI

Secret key generation, generating encrypted format for index of the document, Encrypting original document, Trapdoor key generation, Searching process, decryption of retrieved document these are the modules used in this project. First the key is generated by the random numbers chose by the data owners. This key is used to encrypt the data owner document and index of the document. Data owner first examine the tokenizer and parser for retrieving all the keywords for encrypting their document. And then every document is considered as a vector and splitting the document after applying the dimension expanding method. And then the split index is outsourced in the cloud server. Then the document is encrypted by the symmetric encryption algorithm. AES is the best one for symmetric key encryption. If the data user wants to retrieve the data from the cloud, he will be authorized by the data owner. For that first the data user sends the request to the data owner. After verifying the user's authenticity, the data owner sends the trapdoor key to the data user for accessing the data. By using the trapdoor key, the user sends the query request to the cloud server. Then the server finding the relevance score between the query request and stored documents. If any matched cluster found by the server, extract the encrypted document belongs to the cluster. The top-k extracted document is given to data user.

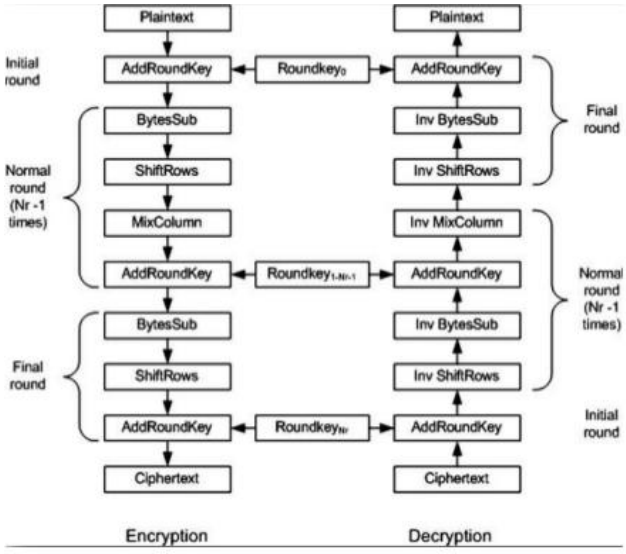

Fig 3 AES Symmetric Key encryption Algorithm

4.3 Dynamic K-Means Algorithm

K-means is the familiar algorithm for clustering process. Initially finding minimum relevance score between the document and center is a vital process in clustering. This process holds the threshold value for the clustering process if threshold value is higher than the found relevance score, all the documents are reassigned and new cluster center is added. This process is continuous till the k value is constant. This process is done at dynamically, so this process is called as Dynamic K-Means Algorithm. For searching Process we use the backtracking algorithm. Backtracking is appealed for problems which grant the perception of a "limited candidate solution" and a relatively fast test of whether it can possibly be finished to a valid solution.

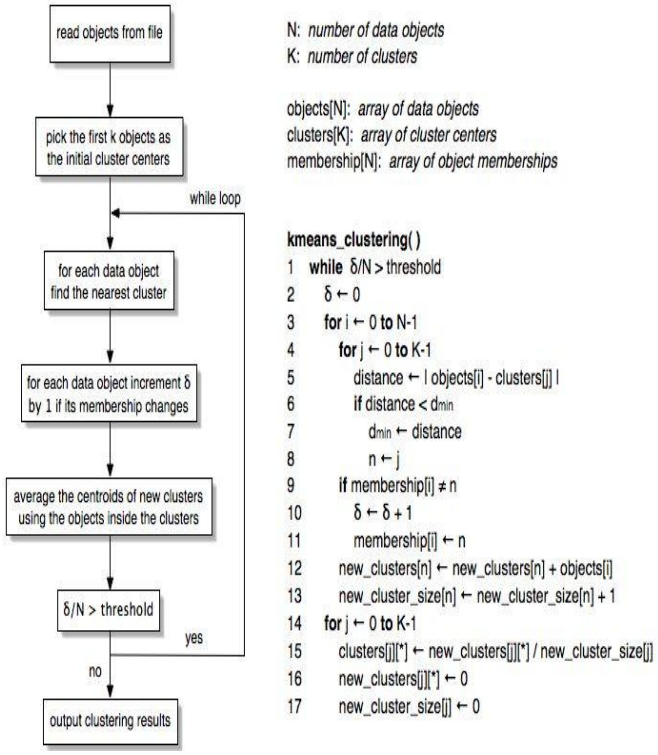

Fig 4 K-Means Clustering Algorithm

Using this clustering algorithm the documents are hierarchically clustered, and the clustered document is accurately retrieved from the cluster by using backtracking algorithm. The hash tree of the cluster document is applied to this algorithm and it processes the linearly ordered documents. And finally get the top-k document based on the query request. The root node of the document hash tree is signature by some cryptographic technique like RSA signature generation and DSA signature generation algorithms. Before that the root node is hashed based on their child node, this root node dictates dataset which contains all the dataset. When outsourcing the hash tree of the document, the signature root node also outsourced in the cloud server. This fig 5 is the format of clustering documents using k-means algorithm.

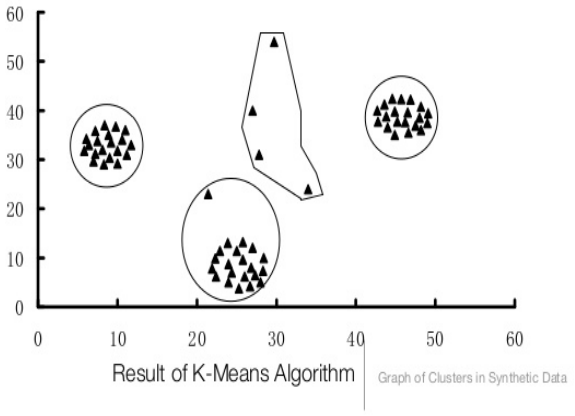

Fig 5 K-means clustering

The simple code for backtracking algorithm is follows.

We can implement a simple recursive algorithm with backtracking:

```
ArrayList<Point> current_path = new ArrayList<Point>();
public static boolean getPaths(int x, int y) {
    Point p = new Point(x, y);
    current_path.add(p);
    if (0 == x && 0 == y) return true; // current_path
    boolean success = false;
    if (x >= 1 && is_free(x - 1, y)) { // Try right
        success = getPaths(x - 1, y); // Free!  Go right
    }
    if (!success && y >= 1 && is_free(x, y - 1)) { // Try down
        success = getPaths(x, y - 1); // Free!  Go down
    }
    if (!success) {
        current_path.remove(p); // Wrong way!
    }
    return success;
}
```

Fig 6 Backtracking algorithm

5. PERFORMANCE AND SECURITY ANALYSIS

Here, we express full experimental results of the proposed technique on real data set:

A. Efficiency. The proposed scheme is characterized in the above sections. In our scheme, we use dynamic k-means algorithm for clustering process. The time of generating key is based on the random number generation. It will not take the much more time to generate keys. And the trapdoor key generation, encrypting document and index creation, searching process and final decryption process are computationally efficient and time complexity is very low. **B. Accuracy.** Using k-means algorithm for clustering process gives an accurate solution for grouping the documents. This accuracy makes the searching process efficient.

C. Privacy. In this process, data owner and data user are trusted parties except cloud server. Cloud server really doesn't know about the original format of the documents. So he is called as semi-trusted third party. So the privacy of data, keyword, and index are retained in this system. For a clear comparison, our proposed scheme gains score higher than the original MRSE. Since the original scheme exploits accurate match, it must loss some similar words which is similar with the keywords. However, our scheme can make up for this disadvantage, and retrieve the most relevant files.

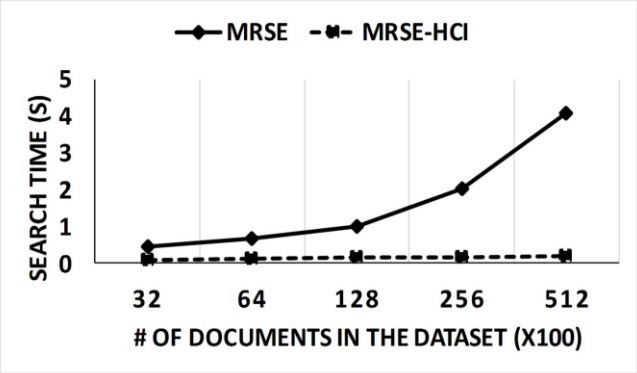

Fig 7 Searching time with an increasing amount of documents

Fig 7 shows that even the number of document is increased to high range; the searching time is increased only linearly. This represent the time complexity is very low to this system. It proves even when the data volume is increased into large, our proposed model reduce the time complexity into logarithmic to search query document.

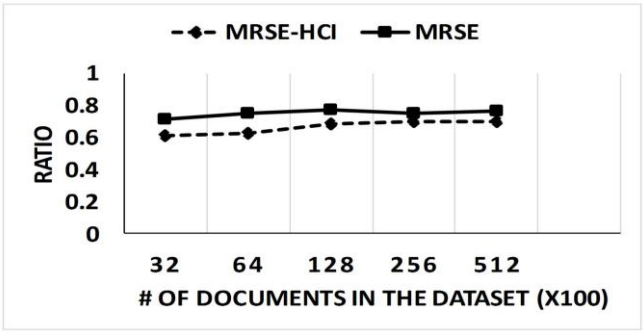

Fig 8 Relevance between documents and query request

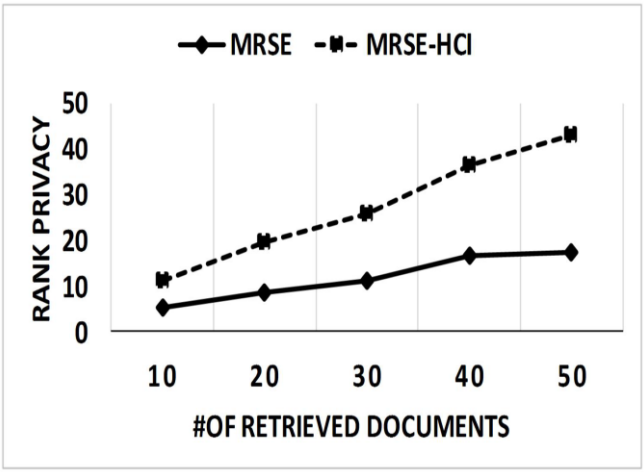

Fig 9 Privacy maintenance

Fig 8 denotes when lot of document store in the database in encrypted format, retaining liaison between the original documents and its encrypted format. If the relationship is retained that time only we can retrieve the target document based on query request. This problem is solved here. This clustering process retain the relationship even huge amount of document will be stored in database

Fig 9 mentions the security privacy of the personal documents. When finding top-k documents, the information may spread out of server. But here the privacy is maintained for ranking system.

These graphical representations are proved by experimental analysis. By this we can prove the efficiency, convenience, security, confidentiality, accuracy, freshness of the proposed system.

6. CONCLUSION

In this paper, as a primary effort, we instigate and solve the problem of advocating effective multi-keyword raked search over encrypted data for improving powerful usage of automatically stored encrypted data in cloud computing. We first propose a basic scheme and show that by existing schemes are inefficient to achieve search query request. So We propose a hierarchical clustering method for group the document. It will be very effective based on k-means clustering algorithm. And the search process is very accurate based on backtracking algorithm. By using this two schema

our system's accuracy, correctness, completeness ands freshness are verified. An experimental platform is erected to evaluate the search efficiency, accuracy, and rank security. The experiment result demonstrates that the proposed architecture not only properly solves the multi-keyword ranked search problem, but also gives an enhancement in search efficiency, rank security, and the relevance between retrieved documents.

7. REFERENCES

[1] 1. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. of IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, March 2010.

[2] A.Singhal, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001

[3] D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data. in Security and Privacy", 2000. S&P 2000, Proceedings 2000 IEEE Symposium, IEEE, (2000).

[4] C. Chen, X. J. Zhu, P. S. Shen, and J. K. Hu, "A Hierarchical Clustering Method For Big Data Oriented Ciphertext Search," presented at Proc. BigSecurity, Toronto, Canada, Apr. 27-May. 2, 2014.

[5] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. of ACNS'05, 2005.

[6] N. Cao, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", INFOCOM, 2011 Proceedings IEEE, IEEE, (2011).

[7] C. Wang, N. Cao, J. Li, K. Ren, and W. J. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," in Proc. ICDCS, Genova, ITALY, 2010.

[8] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," Computers, IEEE Transactions on, vol. 62, no. 2, pp. 362–375, 2013.

[9] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," SIAM Journal on Computing, vol. 32, no. 3, pp. 586–615, 2003.

[10] H. H. Pang, and K. L. Tan, "Authenticating query results in edge computing," in Proc. ICDE, Boston, MA, 2004, pp. 560-571.

[11] H. H. Pang, and K. L. Tan, "Authenticating query results in edge computing," in Proc. ICDE, Boston, MA, 2004, pp. 560-571.